Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

A10

1. (Original) A digital signal processor comprising

two execution pipelines capable of executing RISC instructions;

instruction fetch logic that simultaneously fetches two instructions and routes them to respective pipelines; and

control logic to allow the pipelines to operate independently.

2. (Original) The digital signal processor of claim 1, wherein the instruction fetch logic includes logic that fetches dual SIMD instructions.

3. (Currently Amended) The digital signal processor of claim 1, further including two registers each half the length of a word fetched for memory, and

wherein the instruction fetch logic **[[that]]** fetches a single word into the two registers simultaneously.

4. (Original) The digital signal processor of claim 1, further including an eight port general register file.

5. (Original) The digital signal processor of claim 4, wherein the general register file includes four read registers and four write registers.

6. (Original) A digital signal processor capable of integrating subopcodes into an established instruction set comprising

a memory storing instruction with the new opcodes;

an instruction decoder that identifies a relocatable opcode to designate 64 subopcodes;

and a subopcode detector that decodes subopcodes if the instruction decoder identifies the relocatable opcode.

7. (Currently Amended) A digital signal processor comprising

a register pair; and

hardware configured to execute ~~means for executing~~ a complex multiply instruction on ~~a number~~ complex numbers stored in the register pair, where each complex number includes a real portion and an imaginary portion, the hardware being configured to:

subtract a result of multiplying the imaginary portions of the complex numbers from a result of multiplying the real portions of the complex numbers to produce a real result,

add a result of multiplying the real portion of a first complex number and the imaginary portion of a second complex number to a result of multiplying the imaginary portion of the first complex number and the real portion of the first complex number to produce an imaginary result, and

store the real result and the imaginary result in a register pair. ~~including~~

~~first means for performing multiply instructions on higher order portions of each register in the register pair,~~

~~second means for performing multiply instructions on the remaining portions of each register in the register pair, and~~

~~third means for combining the results from the first and second means.~~

8. (Currently Amended) A circular buffer control circuit comprising

a first number of circular buffer start registers;

a first number of circular buffer end registers, each associated with a different one of the circular buffer start registers; and

circular buffer control logic including

A10

means for comparing a pointer to an **[[and]]** address in a selected one of the circular buffer end registers, and

means for restoring the address in the one of the circular buffer start registers associated with the selected circular buffer end register if the pointer matches the address in the selected circular buffer end register.

9. (Original) A digital signal processor capable of executing zero overhead looping instruction commands comprising:

a register set; and

means for executing a loop instruction command a fixed number of times on a number stored in the register set, including

first means for executing a current instruction stored in a first portion of a first register within the register set;

second means for decrementing a loop count value stored in a second register within the register set;

third means for executing another portion of the current instruction stored in a second portion of the first register and a second register within the register set.

10. (Original) The digital signal processor of claim 9, further including means for exiting the loop instruction command when the loop count value reaches zero.